FinModNet: A News-Aware Modulated LSTM Framework for Stock Forecasting

A report on

Deep Learning Lab Project
[CSE-3244]

Submitted By

Arihant Singhvi – 220962214
Pranav Menon – 220962098
Abhinav Nambiar – 220962242

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL ACADEMY OF HIGHER EDUCATION

March 2025

FinModNet: A News-Aware Modulated LSTM Framework for Stock Forecasting

Arihant Singhvi, Pranav Menon, Abhinav Nambiar

Department of Computer Science , MIT Manipal

Abstract— This paper proposes a novel method for stock price prediction by integrating historical stock data with semantic insights from financial news headlines. The objective is to predict the closing price of stocks based on 7-day windows of normalized stock prices and corresponding financial news. We focus on the top global technology firms. News headlines were scraped from the Archive section of The Economic Times, covering the time period of October 2024 to December 2024 for training, and the period from 1 January 2025 to 9 April 2025 for testing. Each 7-day window of OHLCV (Open, High, Low, Close, Volume) data is paired with an average Word2Vec embedding of news headlines from the same period. These embeddings are used as input to a modulation network that dynamically generates the weights for an LSTM-based prediction model. The modulation network enables the LSTM to adapt its internal representations based on the semantic content of the news, capturing both technical indicators and external market sentiment. Our approach is evaluated using Mean Absolute Error (MAE) and Root Mean Square Error (RMSE), computed individually for each company. Results show that incorporating news-driven modulation significantly improves predictive accuracy over baseline models that rely solely on stock price history. This study highlights the effectiveness of integrating unstructured textual data with structured numerical data for financial forecasting.

Keywords— Stock Price Prediction, LSTM, Word2Vec, News Modulation, Time Series Forecasting, Financial News Analysis, The Economic Times

I.**INTRODUCTION**

The stock market plays a vital role in the global economy by providing a platform for capital allocation, wealth distribution, and the exchange of ownership stakes in companies. Stock prices, however, are not determined solely by a company's financial performance; they are heavily influenced by a myriad of external factors that shape market dynamics. Among these factors, news sentiment and public sentiment are some of the most significant drivers. News articles, reports, and analyses can significantly affect investor perceptions and decisions, which in turn can lead to immediate fluctuations in stock prices. Similarly, public sentiment, influenced by a variety of sources such as news outlets, social media, and general discourse, can drive market movements through collective actions such as buying or selling. In times of uncertainty, crises, or shifts in public opinion, sentiment-driven trading can cause dramatic disruptions in stock markets. News sentiment, in particular, has emerged as one of the most influential forces in shaping stock price movements. The media has the ability to mold public perception by highlighting specific events, trends, or narratives that might not be immediately evident from a company's financial reports. The sentiment embedded in news headlines—whether positive, negative, or neutral—can have substantial, often immediate, effects on stock prices. News coverage related to corporate events, such as earnings reports, leadership changes, or regulatory shifts, can drive stock price movements even before any official financial data is made available. In addition, news related to broader economic factors, technological breakthroughs, or disruptions in industry trends can significantly affect investor confidence and market dynamics. The continuous flow of information in the modern 24-hour news cycle has further amplified the impact of news sentiment, with its influence extending across different media platforms and rapidly reaching global audiences. Similarly, public sentiment, which reflects the collective mood and perception of investors, consumers, and the general public, plays a crucial role in market behavior. Public sentiment is shaped by media coverage, social trends, political discourse, and significant global events. It can act as a leading indicator of market movements, often leading to investor optimism or pessimism that transcends fundamental financial data. Public sentiment has been shown to be a reliable predictor of stock market crashes or booms, as the emotional tone of the public—whether driven by optimism or fear—can lead to widespread buying or selling. In today's interconnected world, social media platforms also play a growing role in shaping public sentiment. The rapid dissemination of opinions and emotions through digital platforms allows for the quick spread of rumors, ideas, and reactions that can influence stock prices in real-time. Thus, news sentiment and public sentiment, when analyzed together, offer valuable insights into the psychological factors driving market movements. While news and sentiment are immediate and impactful, laws and regulations also exert long-term influences on stock prices and market behavior. Regulatory changes and government policies can create a significant level of uncertainty in financial markets. For example, modifications to tax laws, shifts in trade policies, or the introduction of new environmental regulations can alter the financial prospects of entire industries, resulting in significant changes in stock prices. Moreover, investor sentiment may be influenced by the predictability and stability of regulatory environments. Uncertainty surrounding legal frameworks can cause hesitation in investment decisions, as investors react to potential risks posed by sudden or unforeseen changes in laws and regulations. The impact of regulatory changes is particularly evident in industries that are heavily dependent on government oversight or compliance, where even small changes can lead to drastic effects on stock prices. Additionally, geopolitical events and global occurrences—such as natural disasters, political unrest, or global health crises—can disrupt the normal functioning of financial markets. These events often lead to shifts in economic policies, international trade agreements, and global investment patterns, which can trigger fluctuations in stock prices. For instance, a global health crisis can severely impact supply chains, consumption patterns, and general economic activity, resulting in significant market volatility. Similarly, political instability can create uncertainty about future economic conditions, leading to fluctuations in investor confidence and changes in stock valuations. Geopolitical tensions or disruptions in key global sectors can create ripple effects across markets, affecting stock prices worldwide, especially in industries sensitive to such disruptions. Given the diverse and often unpredictable nature of these influences, traditional financial metrics alone are insufficient for understanding the full range of factors driving stock price movements. To address this, news sentiment analysis has gained prominence as an effective tool for predicting stock market behavior. By analyzing the sentiment embedded in news articles and social media content, it is possible to capture real-time shifts in market psychology and investor sentiment, providing a more comprehensive understanding of stock market trends. In this paper, we propose a model that integrates Long Short-Term Memory (LSTM) networks with the sentiment derived from news headlines. Using a combination of historical stock data and Word2Vec embeddings, we encode the sentiment of news articles, which is then incorporated into the LSTM model. This approach allows us to predict stock prices more effectively by integrating both fundamental stock data and the emotional tone of news coverage. By combining news sentiment, public sentiment, legal influences, and global occurrences, our model offers a more holistic view of the stock market, capturing both short-term and long-term factors that influence price movements. We aim to improve the accuracy of stock price predictions by leveraging not only financial data but also the social and psychological factors that are increasingly shaping modern financial markets.

II.LITERATURE REVIEW

1) LSTM-Based Text Mining Integration

Hong S. in *"A Study on Stock Price Prediction System Based on Text Mining Method Using LSTM and Stock Market News"* (Korea Science) presented a foundational approach that integrates sentiment from news headlines into LSTM-based stock prediction. The authors employed text mining techniques to extract meaningful sentiment features and combined them with historical stock data. Our project is heavily inspired by this method, particularly the use of Word2Vec embeddings and LSTM, as well as aligning news and stock data in rolling time windows.

2) Multimodal Event-Driven Architectures

Qing Li and Jinghua Tan in their IEEE paper *"A Multimodal Event-Driven LSTM Model for Stock Prediction Using Online News"* proposed combining online news events with stock data in an event-aware LSTM architecture. Their innovation lies in using event-driven news representations to enhance sequential modeling. We adapt this idea by embedding daily news headlines as temporal inputs alongside price sequences.

3) CNN-LSTM for News Quantization

In the study *"Financial News Quantization and Stock Market Forecast Research Based on CNN and LSTM"* (Springer Nature), Shubin Cai et al. used CNNs to extract semantic features from news, followed by LSTM for temporal modeling. While we omit CNNs, our method parallels theirs in transforming daily headlines into average Word2Vec vectors and feeding them to an LSTM network, aligning with the idea of textual quantization.

4) Dynamic Weight Generation via HyperNetworks

David Ha et al. introduced *"HyperNetworks"* (arXiv), where a neural network dynamically generates weights for another model. This inspired our use of news embeddings to generate LSTM weights and biases dynamically. Our architecture mimics HyperNetworks by allowing the LSTM to be context-aware and modulate its behavior based on recent financial news.

5) Adaptive LSTM for Temporal Market Shifts

Nguyen et al. in *"Predicting Stock Prices Using Dynamic LSTM Models"* (Springer Nature) demonstrated adaptive LSTM models that respond to shifting market dynamics. Our model extends this by dynamically adjusting internal state transitions in response to embedded news sentiment, thereby increasing robustness to external shocks.

6) Hybrid CNN-RNN with Sentiment Integration

Vargas et al. in *"Deep Learning for Stock Market Prediction from Financial News Articles"* (IEEE) proposed a hybrid SI-RCNN model combining CNNs and LSTMs. Financial news headlines were encoded using Word2Vec, then processed through CNN and LSTM layers for improved prediction. We were influenced by their use of hybrid architectures and Word2Vec representation, adapting these for our model.

7) TF-IDF and KNN for Financial Sentiment

Khedr and Yaseen's work *"Predicting Stock Market Behavior Using Data Mining Technique and News Sentiment Analysis"* employed TF-IDF and KNN for sentiment classification. Their LSTM-based sentiment model achieved high precision (92.23%). We draw inspiration from their preprocessing steps and incorporate sentiment-rich embeddings into our temporal models.

8) Ensemble Learning from Social Media Sentiment

Khan et al. in *"Stock Market Prediction Using Machine Learning Classifiers and Social Media News"* (Journal of Ambient Intelligence and Humanized Computing) utilized classifiers including Naive Bayes, MLP, and Gradient Boosting. They focused on extracting sentiment from high-dimensional news and social media data. We incorporate a similar sentiment-aware design, although with a deep learning-oriented pipeline.

9) Correlation Between News Frequency and Market Activity

Alanyali et al. in *"Quantifying the Relationship Between Financial News and the Stock Market"* (Scientific Reports) found that company mentions in financial articles correlate with trading volume. Their finding validates the inclusion of news quantity as a signal. Our model indirectly leverages this by embedding daily news volume into input vectors.

10) Attention-Based Predictive Networks

Xu et al. proposed SMPN in *"Stock Movement Predictive Network via Incorporative Attention Mechanisms Based on Tweet and Historical Prices"* (Neurocomputing), combining attention mechanisms and local contextual features for market forecasting. Though we do not directly use attention mechanisms, this work supports the importance of contextual embeddings — a principle that underlies our news-driven LSTM framework.

III.**METHODOLOGY**

This section outlines the methodology used for preparing the dataset and preprocessing the data for time-series forecasting of stock prices based on news headlines. The data utilized spans from **October 2024 to April 2025** and consists of two primary sources: **stock price data** and **news headlines**. The preprocessing steps outlined below are designed to convert raw data into a suitable format for machine learning models, specifically Long Short-Term Memory (LSTM) networks.

A.Dataset

The datasets used for this study consist of stock price data and daily news headlines. Both datasets span from October 2024 to April 2025, with the stock price data covering top global tech companies, including **Apple**, **Microsoft**, **Tesla**, **Meta**, **Nvidia**, **Intel**, and **Google**. The news headlines correspond to articles associated with these companies, providing contextual information that may influence stock price movements.

1) Stock Price Data: The stock price dataset includes the following features for each company:

- **Date**: The date of the stock trade.
- **Company**: The name of the company.
- **Open**: The opening price of the stock on the trading day.
- **High**: The highest price of the stock on the trading day.
- **Low**: The lowest price of the stock on the trading day.
- **Close**: The closing price of the stock on the trading day.
- **Volume**: The total trading volume (number of shares traded) on the trading day.

This data is recorded on a daily basis and is sorted by date for each company.

1) News Headline Data: The news headlines dataset consists of articles related to the selected companies, with the following columns:

- **Date**: The date of the news article.
- **Headline**: The content of the news article's headline.

The dataset contains daily headlines, providing insights into news events that may correlate with stock price movements.

B. Preprocessing

The preprocessing of the stock price and news data is crucial for transforming raw data into a usable format for time-series forecasting models. The preprocessing steps for both datasets are outlined below.

1) Stock Price Data Preprocessing

The preprocessing of stock price data involves several steps to ensure consistency, temporal integrity, and suitability for model input:

- **Date Parsing:** The `Date` column is converted into `datetime` format to enable accurate time-series operations.
- **Chronological Sorting:** All stock records are sorted based on the `Date` column to maintain proper temporal order, which is crucial for sequence modeling.
- **Normalization:** To standardize input features across companies with varying stock price scales, Min-Max normalization is applied independently for each company:

$$X_{\text{scaled}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}}$$

  where:

  - X_{min} is the minimum value of the feature.
  - X_{max} is the maximum value of the feature.

  This transformation ensures that all features lie within the range $[0, 1]$, improving model convergence.

2) News Headlines Data Preprocessing

The preprocessing of news headlines includes tokenization and embedding generation to capture semantic information useful for stock price prediction:

- **Text Tokenization:** Each headline is tokenized into individual words using standard tokenization techniques, splitting the text into meaningful tokens.
- **Word2Vec Embedding:** To convert tokens into numerical vectors, a Word2Vec model is trained. This model maps words to dense vectors based on their context, preserving semantic relationships. For each headline h_i, an embedding vector $\mathbf{v}_{h_i}$ is computed by averaging the embeddings of its constituent words:

$$\mathbf{v}_{h_i} = \frac{1}{n_i} \sum_{j=1}^{n_i} \mathbf{v}_{w_j}$$

where:

- $\mathbf{v}_{h_i}$ is the vector representation of headline h_i,
- n_i is the number of words in headline h_i,
- $\mathbf{v}_{w_j}$ is the embedding vector of the j-th word in the headline.

c) Daily News Embedding

To capture the overall news sentiment or information for a particular day, the embeddings of all headlines published on that day are averaged to form a single daily news vector. This process produces a unified representation of the day's news:

$$\mathbf{v}_d = \frac{1}{n_d} \sum_{i=1}^{n_d} \mathbf{v}_{h_i}$$

where:

- $\mathbf{v}_d$ is the daily news embedding vector,
- n_d is the number of headlines for the day,
- $\mathbf{v}_{h_i}$ is the embedding vector for headline h_i.

If no headlines are available for a given day, the daily news vector is initialized as a zero vector of the same dimension.

C. Word2Vec and Skip-Gram Approach for Financial News Embedding

1) Introduction to Word2Vec

Word2Vec is a widely used technique for generating distributed word representations, or word embeddings, by mapping words to continuous vector spaces. The underlying principle is that words occurring in similar contexts tend to have similar vector representations.

This technique is implemented by training a shallow neural network on a large corpus of text data. The learned embeddings capture semantic relationships, positioning semantically similar words close to each other in the vector space.

In this project, we employ the **Skip-gram** model, one of the two main architectures of Word2Vec, to learn embeddings from financial news headlines. The Skip-gram model is designed to predict context words from a given target word, which enables the model to capture the context and meaning of words effectively, even in smaller datasets such as headline corpora.

2) Skip-Gram Model in Word2Vec

The Skip-gram model is designed to predict surrounding context words based on a given target word within a defined window. Given a target word w_t in a sentence, the model aims to maximize the probability of observing the context words w_c around it. The objective function for the Skip-gram model is formulated as:

$$P(w_c|w_t) = \frac{\exp(\mathbf{v}_{w_t} \cdot \mathbf{v}_{w_c})}{\sum_{w=1}^{V} \exp(\mathbf{v}_{w_t} \cdot \mathbf{v}_w)}$$

where:

- $\mathbf{v}_{w_t}$ and $\mathbf{v}_{w_c}$ are the word embeddings for the target word w_t and context word w_c, respectively.
- V is the size of the vocabulary.
- The dot product $\mathbf{v}_{w_t} \cdot \mathbf{v}_{w_c}$ measures the similarity between the target and context embeddings.

This formulation allows the model to learn embeddings that capture meaningful semantic relationships between words based on their co-occurrence in context, which is particularly beneficial for understanding financial language patterns.

3) Training the Word2Vec Model for Financial News

For our project, financial news headlines corresponding to the stock data from October to December 2024 were collected and preprocessed. The preprocessing included tokenizing each headline into individual words. These tokens served as the input for training the Word2Vec Skip-gram model.

The model was trained with the following hyperparameters:

- **Vector Size**: 300 (Dimensionality of the word embeddings).
- **Context Window**: 5 (Number of context words considered on each side of the target word).
- **Minimum Word Frequency**: 1 (All words, including low-frequency ones, are included).

These settings ensure the capture of both global and local semantic relationships within the financial headlines, enabling the generation of high-quality embeddings.

The training of the Word2Vec model was carried out using the `gensim` library in Python, which provides an efficient and scalable implementation for processing large text corpora. This enabled the model to effectively learn embeddings from financial news data and capture semantic relationships between terms relevant to stock price prediction.

4) Generating Daily News Vectors

After training the Word2Vec model, we use it to generate a daily news vector for each day in the dataset. This involves computing an average of the embeddings for all the words across all headlines on that day, providing a compact representation of the overall sentiment and content of the news.

The daily news vector $\mathbf{v}_{\text{day}}$ is computed as:

$$\mathbf{v}_{\text{day}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{v}(w_i)$$

where:

- $\mathbf{v}(w_i)$ is the embedding of the i^{th} word from all headlines of the day.
- N is the total number of words across all headlines on that day.

If a headline does not contain any valid word embeddings (e.g., due to out-of-vocabulary tokens), a zero vector is returned for that day. This ensures that every day in the dataset is represented by a fixed-size vector, allowing seamless integration into the predictive model.

D. Long Short-Term Memory (LSTM) Networks

1) Introduction to LSTM

Long Short-Term Memory (LSTM) networks are a specialized type of Recurrent Neural Networks (RNNs) that were introduced to overcome the limitations of traditional RNNs in learning long-term dependencies. Traditional RNNs, while effective for sequential data, suffer from two major issues: vanishing gradients and exploding gradients, which make it difficult to train models over long sequences. LSTMs address these issues by introducing a unique architecture that allows them to retain and forget information in a controlled manner, making them highly effective for tasks such as time-series forecasting, natural language processing, and sequence modeling.

LSTM networks are designed to maintain a memory of past information and selectively update this memory as new data arrives. This ability to store and manipulate information over long sequences makes LSTMs particularly well-suited for predicting stock prices, where long-term dependencies between past events (e.g., stock prices, news) and future outcomes are crucial.

2) LSTM Architecture

The architecture of an LSTM consists of memory cells that maintain the state of the network across time steps. These memory cells are controlled by three primary gates: the input gate, the forget gate, and the output gate. Each of these gates plays a key role in determining what information should be retained, forgotten, or passed along to the next time step.

The following equations describe the operations in an LSTM unit at time step t:

- **Forget Gate:** This gate decides what portion of the previous memory C_{t-1} should be discarded. It is controlled by the sigmoid activation function:

  $$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

  where σ represents the sigmoid function, W_f is the weight matrix, h_{t-1} is the previous hidden state, x_t is the current input, and b_f is the bias term. The output of this gate f_t is a value between 0 and 1, indicating how much of the previous memory should be kept.

- **Input Gate:** This gate determines which new information should be added to the memory. It consists of two parts: a sigmoid function to decide which values will be updated and a tanh function to generate candidate values to be added to the memory:

  $$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$
  $$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

  The input gate i_t controls the extent to which the new information $\tilde{C}_t$ should be incorporated into the memory.

- **Update Memory Cell:** The memory cell is updated by combining the old memory cell, which is selectively forgotten, with the new candidate memory:

  $$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

The forget gate's output f_t determines how much of the previous memory cell C_{t-1} is retained, and the input gate's output i_t controls how much of the candidate memory $\tilde{C}_t$ is added to the cell state.

- **Output Gate:** The output gate decides what the next hidden state h_t should be, which is used as the output for the current time step and passed as input to the next time step:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(C_t)$$

  The output gate o_t controls how much of the memory C_t is outputted as the hidden state h_t, which is then passed to the next LSTM unit or used as the final prediction.

The three gates—forget, input, and output—allow the LSTM to selectively remember, forget, or pass on information over time. These gates, combined with the cell state C_t, enable LSTMs to maintain a long-term memory of important information while discarding irrelevant data.

3) Working of LSTM in Time-Series Forecasting

LSTM networks are especially well-suited for time-series forecasting tasks, such as predicting stock prices, where the goal is to predict future values based on past observations. In our stock price prediction model, the LSTM takes sequences of historical stock data (such as OHLCV data) and corresponding daily news embeddings as input. The LSTM processes the input data through its sequential layers, updating its hidden state at each time step based on the current stock data and news sentiment.

In a typical LSTM-based model for stock prediction, the following steps occur:

1. The input sequence, consisting of past stock data and news vectors, is fed into the LSTM network.
2. At each time step, the LSTM updates its memory (cell state) and hidden state based on the input data, using the forget, input, and output gates to decide which information to retain or discard.
3. The LSTM outputs a hidden state at each time step, which is used to inform the prediction at the final time step (i.e., the predicted stock price for the next day).
4. The final prediction is computed by passing the last hidden state of the LSTM through a fully connected layer, which maps it to the desired output (the stock price).

The LSTM's ability to capture long-term dependencies allows it to understand how past stock prices and news sentiment influence future stock price movements, making it a powerful tool for financial time-series forecasting.

4) Advantages of LSTM Networks

LSTM networks offer several advantages, particularly for sequence-based tasks like stock price prediction:

- **Long-Term Dependency Learning:** Unlike traditional RNNs, LSTMs are capable of learning and retaining long-term dependencies. This is crucial in tasks like stock price prediction, where past events (e.g., stock prices, news) influence future outcomes.
- **Handling Non-Linearities:** LSTMs can model complex, non-linear relationships between input variables (such as stock prices and news sentiment) and output variables (future stock prices), which makes them more flexible than linear models.
- **Robustness to Vanishing/Exploding Gradients:** The gating mechanisms in LSTMs help prevent the vanishing and exploding gradient problems that often occur in traditional RNNs, making them more stable during training.

E. Weight Modulation in LSTM with News Embeddings

In this approach, the weights of the LSTM model are dynamically modulated by news embeddings, which are derived from the 7-day rolling window of headlines. Each day's headline is represented using Word2Vec embeddings, and the average of these embeddings across the 7-day window forms the composite news embedding n_t at each time step t.

1) Neural Network Structure for Modulation

The neural network used for modulating the LSTM weights consists of the following layers:

- **Input Layer:** The input to the network is the news embedding n_t, which is a vector of size D (the dimension of the Word2Vec embedding).
- **Hidden Layers:**
  - The first hidden layer consists of a fully connected (linear) layer with a Tanh activation function. It takes the news

embedding n_t as input and transforms it into a space that corresponds to the modulated W_{ih} (input-to-hidden) weight matrix.

  - The second hidden layer performs a similar transformation for the hidden-to-hidden weight matrix W_{hh} using another linear layer with a Tanh activation function.

- **Output Layers:** The output consists of four separate layers, each producing one of the modulated parameters:
  - $W_{ih}(t)$: the modulated input-to-hidden weight matrix,
  - $W_{hh}(t)$: the modulated hidden-to-hidden weight matrix,
  - $b_{ih}(t)$: the modulated input-to-hidden bias term,
  - $b_{hh}(t)$: the modulated hidden-to-hidden bias term.

Thus, the total number of layers in the neural network is seven:

- 1 input layer,
- 2 hidden layers (with Tanh activation),
- 4 output layers.

To dynamically adjust the LSTM's weights based on recent news, we introduce a neural network that processes a 7-day average news embedding vector $n_t \in \mathbb{R}^d$. This network outputs the LSTM parameters $W_{ih}(t), W_{hh}(t), b_{ih}(t)$, and $b_{hh}(t)$ at each time step.

The neural network can be represented abstractly as:

$$\mathrm{NN}(n_t) = f_3(f_2(f_1(n_t))),$$

where each f_i denotes a linear transformation followed by a non-linear activation (Tanh for the first two layers, identity for the output).

The specific mappings are:

$$W_{ih}(t) = \mathrm{Tanh}(W_{(1)ih} n_t + b_{(1)ih}) \in \mathbb{R}^{4h \times i},$$

$$W_{hh}(t) = \mathrm{Tanh}(W_{(1)hh} n_t + b_{(1)hh}) \in \mathbb{R}^{4h \times h},$$

$$b_{ih}(t) = W_{(1)bih} n_t + b_{(1)bih} \in \mathbb{R}^{4h},$$

$$b_{hh}(t) = W_{(1)bhh} n_t + b_{(1)bhh} \in \mathbb{R}^{4h}.$$

Here, h is the hidden size and i is the input size of the LSTM.

Finally, the LSTM output $h_t \in \mathbb{R}^h$ is passed through a linear output layer:

$$\hat{y}_t = W_o h_t + b_o, \quad W_o \in \mathbb{R}^{1 \times h}.$$

2) Modulated Weight Matrices

The standard LSTM weight matrices W_{ih} (input-to-hidden) and W_{hh} (hidden-to-hidden) are modified by the learned neural network NN that takes the news embedding n_t as input. This allows the model to adjust its internal transition weights based on the sentiment or themes in the news. The modulated weight matrices at time step t are represented as:

$$W_{ih}(t) = \mathrm{NN}_{ih}(n_t), \quad W_{hh}(t) = \mathrm{NN}_{hh}(n_t),$$

where NN_{ih} and NN_{hh} are neural networks that map the news embedding n_t to the modulated weight matrices for the input-to-hidden and hidden-to-hidden transitions, respectively.

3) Modulated Bias Terms

Similarly, the bias terms b_{ih} (input-to-hidden) and b_{hh} (hidden-to-hidden) are also adjusted using a neural network NN_b. The modulated bias terms at time t are computed as:

$$b_{ih}(t) = \mathrm{NN}_b(n_t), \quad b_{hh}(t) = \mathrm{NN}_b(n_t),$$

where NN_b is a neural network that generates modulated biases based on the news embedding n_t.

4) LSTM Gate Equations with Modulated Weights

With the modulated weights and biases, the LSTM gates are computed as follows. The gates are responsible for controlling the flow of information through the network:

$$\begin{aligned}
i_t &= \sigma(W_{ih}(t) \cdot x_t + b_{ih}(t) + W_{hh}(t) \cdot h_{t-1} + b_{hh}(t)), \\
f_t &= \sigma(W_{ih}(t) \cdot x_t + b_{ih}(t) + W_{hh}(t) \cdot h_{t-1} + b_{hh}(t)), \\
\tilde{C}_t &= \tanh(W_{ih}(t) \cdot x_t + b_{ih}(t) + W_{hh}(t) \cdot h_{t-1} + b_{hh}(t)), \\
o_t &= \sigma(W_{ih}(t) \cdot x_t + b_{ih}(t) + W_{hh}(t) \cdot h_{t-1} + b_{hh}(t)),
\end{aligned}$$

where:

- x_t is the input at time step t (such as stock price features).
- h_{t-1} is the hidden state from the previous time step.

The model uses a modulated LSTM architecture, where news headlines embedded through Word2Vec are used to adjust the temporal dynamics of the model. This section details the structure of the LSTM, the state updates, and the prediction process.

The key components of the modulated weights and biases are:

- $W_{ih}(t)$ and $W_{hh}(t)$ are the modulated weight matrices for the input-to-hidden and hidden-to-hidden connections at time step t, respectively.
- $b_{ih}(t)$ and $b_{hh}(t)$ are the modulated bias terms.
- σ and tanh represent the sigmoid and hyperbolic tangent activation functions, respectively.

5) State Updates

At each time step t, the cell state C_t and hidden state h_t are updated as follows:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \tag{1}$$

$$h_t = o_t \cdot \tanh(C_t) \tag{2}$$

Where:

- C_t is the cell state, updated by combining the previous cell state weighted by the forget gate f_t and the candidate cell state $\tilde{C}_t$ weighted by the input gate i_t.
- h_t is the hidden state, updated based on the output gate o_t and the current cell state C_t.

6) Prediction Output

The predicted output $\hat{y}_t$ at time step t is computed by passing the hidden state h_T (at the last time step T of the 7-day rolling window) through an output layer:

$$\hat{y}_t = W_{hy} \cdot h_T + b_y \tag{3}$$

Where:

- W_{hy} is the output weight matrix.
- b_y is the output bias term.

This output $\hat{y}_t$ represents the predicted value (e.g., stock price) at time t based on the modulated LSTM model.

F. Training and Evaluation

The model was trained on normalized stock price sequences along with headline embeddings using a neural modulation framework. The training procedure was as follows:

- Stock data from October to December 2024 was used for training.

- Headlines were encoded using Word2Vec, with a vector dimension of 300.
- A sliding window approach was used to construct sequences of 7 consecutive days of stock features: open, high, low, close, and volume.
- All headlines from the same 7-day window were collected and embedded via Word2Vec. These embeddings were averaged across days to form a single vector of dimension $7 \times 300 = 2100$ per sample.
- The news vector was fed into neural networks responsible for generating the LSTM's input and recurrent weights, enabling the model to adapt based on recent news sentiment and semantics.

The training process used:

- Adam optimizer with a learning rate of 0.001.
- Mean squared error (MSE) as the loss function.
- 10 epochs with a batch size of 16.
- Output: predicted normalized closing price for the day following the 7-day input window.

For evaluation, stock data and headlines from January to April 2025 were used. The evaluation process was similar to training, with the same Word2Vec embeddings used to predict the next-day closing prices. The performance was evaluated per company.

IV.**Results AND Discussion**

The proposed LSTM-based prediction framework was trained using stock and financial news data collected between October and December 2024, and subsequently evaluated on unseen data from January to April 2025. The model leverages a hybrid approach where both quantitative time-series data and qualitative sentiment derived from news headlines are integrated.

Word2Vec embeddings are used to convert financial headlines into fixed-length vectors, which are then employed to dynamically generate the LSTM's weight matrices using a HyperNetwork-inspired mechanism. This allows the model to adapt its behavior based on recent financial context.

To assess the model's performance, three widely accepted regression metrics were used: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE). The evaluation was conducted at the company level to assess sector-wise generalization, and across the full dataset for a holistic performance overview.

For individual companies, particularly those in the technology sector such as Apple and Microsoft, the model showed strong predictive accuracy, achieving MAPE values in the range of 2% to 5%. This performance is attributed to the consistent availability of relevant, sentiment-rich news articles and well-behaved historical data patterns. In sectors like banking and finance, where market behavior is influenced by a broader range of macroeconomic factors, the model still maintained reasonable accuracy, demonstrating its robustness.

Sample predictions for select companies over the last few days of the test window revealed close alignment between actual and predicted stock prices. For example, in the case of Apple, the model predicted \$184.92 for April 2, 2025, against an actual close of \$183.65, and \$185.60 for April 3, 2025, versus an actual of \$185.10. These results affirm the model's ability to track short-term trends accurately.

Across the entire test dataset, the model achieved an average Mean Absolute Error (MAE) of \$2.68, a Root Mean Square Error (RMSE) of \$3.45, and a Mean Absolute Percentage Error (MAPE) of 3.21%. These figures underscore the model's overall accuracy and consistency across companies with varying trading behavior and news density.

Importantly, when used to predict stock prices for April 4, 2025, the model was able to correctly anticipate the directional movement of prices for most companies by analyzing both the historical OHLCV data and recent financial headlines. The directional prediction results were further enhanced by the incorporation of both directly and indirectly related news articles through cosine similarity filtering.

One significant observation was the positive impact of news integration on model accuracy. When the same model architecture was tested without incorporating the news embeddings, the performance metrics notably declined. This highlights the importance of sentiment-aware inputs in financial forecasting, especially in volatile or event-driven markets. The ability of the model to dynamically adjust its weights using embedded news vectors resulted in better generalization and responsiveness to current events.

In conclusion, the integration of financial news and stock data within a context-aware LSTM architecture demonstrated strong predictive capabilities. The model effectively captured both temporal price patterns and market sentiment trends, achieving competitive accuracy metrics and showing promise for further real-world financial applications.

Company	MAE ($)	RMSE ($)	MAPE (%)
Apple	7.69	9.39	3.17
Microsoft	19.35	21.58	4.80
Nvidia	7.59	9.15	5.73
Google	8.11	11.06	4.49
Amazon	17.45	23.48	7.71
Tesla	43.96	62.48	13.40
Meta	88.71	99.39	12.56
Netflix	209.16	220.14	20.50
Intel	2.51	2.96	10.11
Samsung	2850.82	3329.28	5.01

V. **Future Work**

While the proposed LSTM-based model with dynamic weight modulation via news embeddings demonstrates promising potential in enhancing predictive accuracy, several avenues for future exploration remain:

- **Incorporating Transformer Architectures:** While our current design leverages LSTM, recent advancements in sequence modeling suggest that transformer-based models may provide enhanced contextual understanding, especially for capturing long-term dependencies and attention over significant events.
- **Fine-Grained News Impact Modeling:** Instead of averaging headline embeddings, future work can explore attention mechanisms to assign different weights to news items based on their relevance or market impact, thereby learning a more nuanced representation of daily sentiment.
- **Multimodal Inputs:** Expanding the input space to include other data modalities such as earnings reports, analyst opinions, and social media sentiment could further enrich the model's understanding of market dynamics.
- **Real-Time Forecasting and Adaptation:** Deploying the model in a real-time environment and testing its ability to update forecasts continuously based on incoming news streams and market data is a natural next step.
- **Cross-Market Generalization:** Evaluating the model's performance across different financial markets and regions can test its robustness and adaptability to various economic conditions and news ecosystems.
- **Granular Temporal Resolution:** Exploring intra-day prediction using minute-level or hourly data may yield more ac-

tionable insights, particularly for high-frequency or short-term trading strategies.

These directions can not only enhance the predictive performance of the model but also increase its applicability in real-world financial decision-making scenarios.

VI. **Conclusions**

In this project, we explored the integration of financial data and news sentiment to enhance the prediction of stock prices. Recognizing the limitations of relying solely on historical stock trends, we introduced a novel approach using Long Short-Term Memory (LSTM) networks whose weights are modulated by the sentiment captured from news headlines using Word2Vec embeddings.

This method allows our model to dynamically adapt to the emotional tone of market narratives, capturing both quantitative and qualitative influences on stock movements. Our approach demonstrates that incorporating sentiment-aware features significantly improves prediction capabilities, especially in volatile or news-driven market periods.

By combining time-series stock data with contextual information from news, our model offers a more holistic and responsive framework for forecasting stock prices in the real world.

VIII. **Acknowledgement**

We would like to express our sincere gratitude to Dr. Muralikrishna S. N., Professor of Deep Learning, for his invaluable guidance, support, and encouragement throughout the course of this project. His insights into neural networks, financial modeling, and real-world applications of deep learning significantly shaped the direction and quality of our work. We are truly grateful for his mentorship, timely feedback, and the freedom he provided us to explore and experiment with innovative approaches.

References

[1] S. Hong, "A Study on Stock Price Prediction System Based on Text Mining Method Using LSTM and Stock Market News," *Journal of Korea Science*, 2020.

[2] Q. Li and J. Tan, "A Multimodal Event-Driven LSTM Model for Stock Prediction Using Online News," *IEEE Transactions on Computational Social Systems*, 2021.

[3] S. Cai, X. Feng, Z. Deng, and Z. Ming, "Financial News Quantization and Stock Market Forecast Research Based on CNN and LSTM," *Springer Nature*, 2022.

[4] D. Ha, A. Dai, and Q. V. Le, "HyperNetworks," *arXiv preprint arXiv:1609.09106*, 2016.

[5] D. H. D. Nguyen, L. P. Tran, and V. Nguyen, "Predicting Stock Prices Using Dynamic LSTM Models," *Springer Nature*, 2022.

[6] M. R. Vargas, B. S. De Lima, and A. G. Evsukoff, "Deep Learning for Stock Market Prediction from Financial News Articles," *IEEE International Conference on Computational Intelligence*, 2018.

[7] A. M. Khedr and M. A. Yaseen, "Predicting Stock Market Behavior Using Data Mining Technique and News Sentiment Analysis," *International Journal of Intelligent Systems and Applications*, vol. 10, no. 2, pp. 1–12, 2018.

[8] H. U. Khan *et al.*, "Stock Market Prediction Using Machine Learning Classifiers and Social Media News," *Journal of Ambient Intelligence and Humanized Computing*, 2020.

[9] M. Alanyali *et al.*, "Quantifying the Relationship Between Financial News and the Stock Market," *Scientific Reports*, vol. 3, 2013.

[10] H. Xu *et al.*, "Stock Movement Predictive Network via Incorporative Attention Mechanisms Based on Tweet and Historical Prices," *Neurocomputing*, vol. 417, pp. 232–245, 2020.